4️⃣

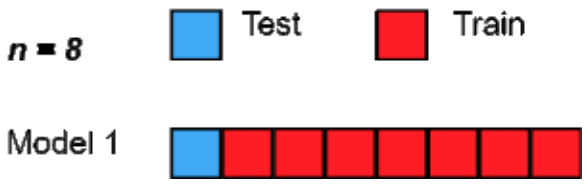
Evaluation

Course	RecSys
Files	Lecture2.4 - Evaluation-2.pdf
Link	
Status	Done
Type	Lecture

Offline Evaluation

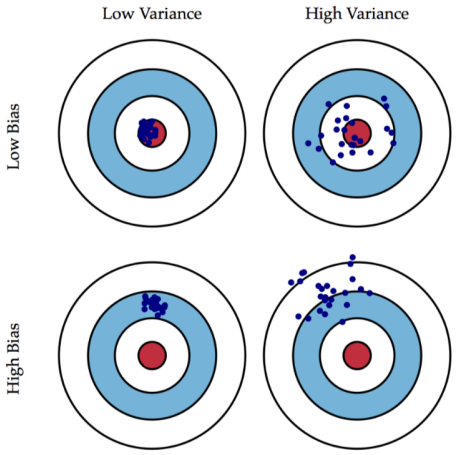
Data

- How to split training vs testing sets?
  - **Holdout**: take out a random percentage of ratings
    - Risk of overfitting as tested users are not totally unknown to the model
  - **N-fold cross validation**: split users into partitions (10 for example) and take 9 for training and 1 for testing
  - **Leave p-out cross-validation**: exhaustive (meaning all possibilities are taken into account) as the image below shows, it is done in iterations and in each iteration a observation is left out and the model is trained against the rest and then tested against the observation that was left out, keeps repeating until every observation is left out once.
    I believe an observation is for example a user or item and not each rating, but I am not sure



    - **Bootstrap**: also works in iterations, take out x% to test with, train with the rest, test with the x%, sample again and repeat, when sampling again we put the x% back so we actually sample from all the data

- Data has bias and variation, our analysis can only be as good as the variance and bias in the data



Metrics

Blue highlight is important

- **Accuracy of predictions - "raw numbers"**

- **Mean Absolute Error (MAE)**
  - computes the deviation between predicted ratings and actual ratings

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - r_i|$$

- **Normalized Mean Absolute Error (NMAE)**
  - same as before but the returned values are between 0 and 1

$$NMAE = \frac{MAE}{r_{max} - r_{min}}$$

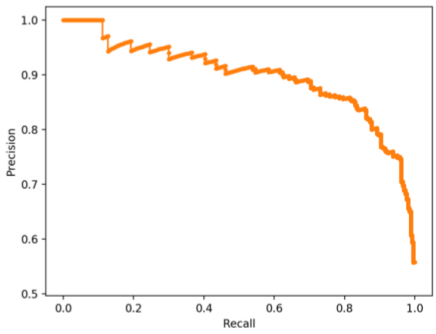
- **Root Mean Square Error (RMSE)**
  - is similar to MAE, but places more emphasis on larger deviation

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2}$$

- Accuracy of classifications - "**Confusion Matrix**"

		Reality	
		Actually Good	Actually Bad
Prediction	Rated Good	True Positive (TP)	False Positive (FP)
	Rated Bad	False Negative (FN)	True Negative (TN)

- **Precision vs Recall**
  - **Precision** = good recommendations/all recommendations
  - **Recall** = good movies recommended/all good movies
  - Usually there is a trade off between precision and recall, if one increases the other decreases and vice versa. Essentially you are recommending better items but you are not recommending all to have a better precision, and vice versa



- Precision and recall can also be calculated for specific set of items: **precision@k / recall@k**
- **F1** combines recall and precision into a singular value

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

there is a variation where recall has a weight called

$$F_\beta$$

- **Hit rate** 1 if at least one item from the user profile is recommended, 0 otherwise

- Ranks metrics - not only if the correct items are recommended but also are they in the correct order

$$rankscore = \frac{rankscore_p}{rankscore_{\max}}$$

$$rankscore_p = \sum_{i \in h} 2^{-\frac{rank(i)-1}{\alpha}}$$

$$rankscore_{\max} = \sum_{i=1}^{|T|} 2^{-\frac{i-1}{\alpha}}$$

Where:

- h is the set of correctly recommended items, i.e. hits
- rank returns the position (rank) of an item
- T is the set of all items of interest
- α is the ranking half life, i.e. an exponential reduction factor

- **Discounted Cumulative Gain (DCG)**, for a list
  - Logarithmic reduction factor

$$DCG_{pos} = rel_1 + \sum_{i=2}^{pos} \frac{rel_i}{\log_2 i}$$

Where:
- *pos* denotes the position up to which relevance is accumulated
- *reli* returns the relevance of recommendation at position i

- **Idealized Discounted Cumulative Gain (IDCG)**
  - Assumption that items are ordered by decreasing relevance

$$IDCG_{pos} = rel_1 + \sum_{i=2}^{|h|-1} \frac{rel_i}{\log_2 i}$$

- **Normalized Discounted Cumulative Gain (nDCG)**
  - Normalized to the interval [0..1]

$$nDCG_{pos} \frac{DCG_{pos}}{IDCG_{pos}}$$

- User coverage (Ucov)
  - Measures capability of the system to compute recommendations for a large share of the population

$$Ucov = \frac{\sum_{u \in U} \rho_u}{|U|}$$

- Catalog coverage (Ccov)
  - reflects the total share of items that are recommended to at least one user

$$Ccov = \frac{|\bigcup_{u \in U} recset_u|}{|I|}$$

- Intra-list similarity (ILS)
  - measure of the diversity of recommendation lists

$$ILS_u = \frac{\sum_{i \in recset_u} \sum_{j \in recset_u, i \neq j} sim(i, j)}{2}$$

- In offline evaluation there is no information about false negatives (Rated bad, but actually good)